



Label Space Reduction in MPLS Networks: How Much Can A Single Stacked Label Do?

Solano, Fernando; Stidsen, Thomas K.; Fabregat, Ramon; Marzo, Jose Luis

Published in:
I E E E - A C M Transactions on Networking

Link to article, DOI:
[10.1109/TNET.2007.912382](https://doi.org/10.1109/TNET.2007.912382)

Publication date:
2008

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Solano, F., Stidsen, T. K., Fabregat, R., & Marzo, J. L. (2008). Label Space Reduction in MPLS Networks: How Much Can A Single Stacked Label Do? *I E E E - A C M Transactions on Networking*, 16(6), 1308-1320.
<https://doi.org/10.1109/TNET.2007.912382>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Label Space Reduction in MPLS Networks: How Much Can A Single Stacked Label Do?

Fernando Solano, *Student Member, IEEE*, Thomas Stidsen, Ramon Fabregat, and Jose Luis Marzo

Abstract—Most network operators have considered reducing LSR label spaces (number of labels used) as a way of simplifying management of underlying Virtual Private Networks (VPNs) and therefore reducing operational expenditure (OPEX). The IETF outlined the *label merging* feature in MPLS—allowing the configuration of MultiPoint-to-Point connections (MP2P)—as a means of reducing label space in LSRs. We found two main drawbacks in this label space reduction scheme: *a)* it should be separately applied to a set of LSPs with the same egress LSR—which decreases the options for better reductions, and *b)* LSRs close to the edge of the network experience a greater label space reduction than those close to the core. The latter implies that MP2P connections reduce the number of labels asymmetrically.

In this article we propose a solution to these drawbacks achieved by stacking an additional label onto the packet header. We call this type of reduction Asymmetric Merged Tunnels (AMT). A fast framework for computing the optimal reduction using AMTs is proposed. Our simulations show that the label space can be reduced by up to 20% more than when label merging is used.

Index Terms—Label merging, label space reduction, label stacking, MPLS, multipoint-to-point.

I. INTRODUCTION

MULTI PROTOCOL LABEL SWITCHING (MPLS) is a circuit-oriented technology used for Traffic Engineering (TE). MPLS aims at working with TE schemes by setting up Label Switched Paths (LSPs) as needed to transmit customer's flows efficiently, considering their QoS requirements. In order to satisfy these needs, network operators have to increase their transport capacities keeping both CAPITAL EXPENDITURES (CAPEX) and OPERATIONAL EXPENDITURES (OPEX) as low as possible. Although the forwarding plane capacity has drastically increased in the last years, the control plane has not evolved that fast. In particular, the new tendency of using distributed control planes had led to an increase in OPEX due to the complexity of the monitoring software [1].

Manuscript received December 22, 2005; revised June 7, 2007. First published February 22, 2008; current version published December 17, 2008. Approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor H. Schulzrinne. The work was supported in part by the Department of Universities, Research and Information Society (DURSI) of the Government of Catalonia and the European Social Funds under Grants 2005-SGR-00296, 2006-BE-00272, and 2006-FIR-00109, in part by the Spanish Science and Technology Ministry under M2R3-TEC2006-03883/TCM, and in part by the European COST 293 project.

F. Solano was with the Broadband and Distributed Systems Group, University of Girona, 17071 Girona, Spain. He is now with the Institute of Telecommunications, Warsaw University of Technology, 00-661 Warsaw, Poland (e-mail: fs@tele.pw.edu.pl).

R. Fabregat and J. L. Marzo are with the Broadband and Distributed Systems Group, University of Girona, 17071 Girona, Spain (e-mail: ramon.fabregat@udg.es; joseluis.marzo@udg.es).

T. Stidsen is with the Informatics and Mathematical Modeling Department, Danish Technical University, DK-2800 Kgs. Lyngby, Denmark (e-mail: tks@imm.dtu.dk@udg.es).

Digital Object Identifier 10.1109/TNET.2007.912382

In MPLS, the more LSPs are handled, the more complex the control plane becomes. The complexity required for managing the control plane is proportional to the number of flows and, in the case of MPLS, it can be estimated by the number of labels used (*viz.* label spaces). As a matter of fact, many network operators (e.g., Bell [2], AT&T [3], NEC Japan [4], NEC USA [5]) aim at reducing network nodes label spaces not only because they are a finite resource, but also because this simplifies network management (especially when VPNs are considered) and therefore reduces OPEX.

Moreover, an important motivation concerns the evolution towards a robust Optical Transport Network (OTN). In an effort to provide a fast control plane in OTNs, MPLS basic functionality has been implemented directly in hardware using completely optical devices. In particular, the LASAGNE project [6] aims at developing an All-Optical Label Swapping (AOLS) architecture. An AOLS router needs a special all-optical device (*viz.* an optical label correlator) for each label used in the router. This makes the cost of an AOLS router (and its size) proportional to the number of labels used [7]. In this case, reducing label spaces could lead not only to reducing OPEX, but also to a significant CAPEX reduction.

A. The Label Space Reduction Problem

Once an LSP is established, all the involved LSRs should each use a label in order to identify the LSP. LSP packets are marked with a label that associates a traffic flow with their LSP in the LSR (labels are local to the LSR). When an LSR receives a packet, the LSR looks for the packet label and then searches for a Next Hop Label Forwarding Entry (NHLFE) in its memory that refers to this incoming label. An NHLFE provides information about which interface should be used to reach the next hop in the network [8]. Clearly, the more LSPs an LSR supports, the more NHLFEs are needed.

The method proposed in this paper is based on MPLS label stacking. Label stacks were originally used for LSP tunneling across different MPLS domains. LSP tunneling is a feature that allows us to aggregate a set of LSPs to form a single LSP. To support LSP tunneling and forwarding, the Internet Engineering Task Force (IETF) in [8] and [9] standardized a set of operations that can be performed in MPLS packet header stacks. The following operations can be done *after* the next hop has been computed:

- SWAP: replace the label at the top by a new one,
- PUSH: replace the label at the top by a new one and then push one or more onto the stack, and
- POP: remove the label at the top of the label stack

Each NHLFE associates an incoming label with one of these operations and an outgoing forwarding port. In this way, LSRs

can decide where to forward packets marked with a specific incoming label and, at the same time, set the label stack properly for downstream LSRs. Due to performance issues, the IETF standard imposes considering only the label at the top of the stack. Hence, the forwarding decision is only based on the top label.

Taking advantage of the different possible operations an NHLFE may have, the number of labels used—or label space¹—can be reduced depending on how the NHLFEs are configured, as explained in further sections. When one label is used to forward more than one LSP, it is said that there is a LAbel SPaCe REDuction (LASPARED). The general LASPARED problem can be formulated as:

How can the NHLFEs be set up for a set of LSRs in a network so that the total number of labels used in the network is minimized?

Note that as the number of incoming labels is reduced, the number of NHLFEs is reduced as well.

Although the main motivation for reducing label spaces are due to OPEX savings in classical MPLS networks and CAPEX reduction in AOLS networks, some other reasons and motivations are presented below:

- some protection mechanisms duplicate the label space sizes, see [10] and [11] (i.e., fast failure protection, 1 + 1);
- allowing multicast on MPLS could increase the number of labels used;
- each label must be encoded in a 20-bit field [8], allowing only 2^{20} (a little more than a million labels) different possible labels in an LSR: a large, but finite number;
- although MPLS architecture was designed for fast forwarding, large NHLFE tables can cause long delays while an LSR is looking up the next hop LSR in its forwarding table each time a packet is received [5], [12]–[14];
- the emergence of new protocols and technologies based on MPLS is encouraging researches to consider smaller label spaces. For instance, such as the Ethernet LSPs project [15] and the LASAGNE project [6], mentioned before.

B. Assumptions

In our approach, the following assumptions are considered:

- *About stack sizes.* In order to reduce packet overhead and management complexity, we simplified our method to one pushed label (in addition to those already used for LSP/VPN setup). Another good reason to limit the number of pushed labels in this study is due to the fact that we foresee AOLS router technology with a bound in the number of labels that can be pushed, due to optical synchronization and bandwidth wastage.
- *About FECs.* As many other works on LASPARED (e.g., [3], [4], and [5]), it is assumed that all given LSPs have equivalent FECs (Forwarding Equivalence Class) as recommended in the MPLS architecture [8]. Otherwise, LSP flows cannot be aggregated.
- *About re-routing.* Applegate *et al.* in [3] considered minimizing the label space as a criterion when finding LSP routes. Since ISP customer needs may be measured as QoS

¹We will use these two terms indistinctively: “number of labels used” and “label space.”

requirements, which do not include the label space, we believe that the number of labels should not be regarded as a minimizing objective in any problem that deals with LSP routing; it may be considered at most as a model restriction. Therefore, our work aims at optimizing the number of labels used without concerning with routing computations.

C. Contents and Contributions

In this subsection we present the organization of this article together with a summary of its contributions.

In Section II, the MP2P basic concepts, the label MERGING PROBLEM and its most important contributions are summarized. Then, in Section III, we propose an enhancement for the MP2P reduction by means of a single label stacked. We call this type of reduction *Asymmetric Merged Tunnels* (AMT). Our main contribution in this paper is to study how much the label space can be reduced by using AMTs instead of MP2Ps. Although MP2P was proposed by IETF and has been analyzed by other authors in previous work, the maximum number of labels that can be reduced using MP2P had not been *accurately* computed but only estimated. Therefore, our contributions are focused in computing the maximum number of labels reduced not only with AMT, but with MP2P as well.

Since considering multicast connections in MPLS increases the number of labels used, this particular case is discussed in Section III.C.

We initially thought of using a single optimization model for minimizing label spaces using AMTs (called the *BF-model* and presented in Section IV); however, this solution would generally be too expensive in terms of computational time/space-complexity. Therefore, we focus not only on an optimization model, but also on a *fast* way for carrying it out. For this, we delved into the problem and propose a 2-step framework, called the *Decompose and Match Framework* (D&M). This framework is discussed and justified in detail in Section V.

A set of simulations comparing the maximum reductions of AMTs with respect to MP2Ps are shown in Section VI, and further conclusions and further studies are discussed at the end of the article.

It should be pointed out that no changes in the current MPLS specification are needed to fulfill accurate behavior for all the presented methods in this paper.

II. LABEL MERGING OR MULTIPOINT-TO-POINT FOR LABEL SPACE REDUCTION

The IETF proposed *labels merging* as a way of reducing the label space. This is performed by assigning the same outgoing label to many LSPs, if they share the path to an egress LSR.²

Let $\mathcal{N} = \{\alpha_0, \alpha_1, \dots, \alpha_N\}$ be the set of all LSRs in an MPLS network. Let $\mathcal{P} = \{p_0, p_1, \dots, p_n\}$ be a set of LSP path indexes, i.e., p_i is an index for an LSP route. Let $f_{\alpha_j} : \mathcal{P} \rightarrow \mathcal{N}$ be the routing function for node α_j : given an LSP p_i , the function returns the next hop (downstream to α_j) of the route.

²At the time of publication, the present work does not take into account the correction stated by the authors in [16]. The correction simplifies the way in which MP2P connections are computed and, moreover, improves the reduction in MP2P by 1% in the 40% of the scenarios by discarding non-practical network considerations.

The idea behind label space reduction is to minimize the domain of every function f_{α_j} , i.e., the label space for α_j . In the case of *label merging*, the domain of the function can be replaced by MP2P identifiers. Let $\mathcal{T} = \{t_0, t_1, \dots, t_T\}$ be a set of MP2P indexes. Similarly to f_{α_j} , we can construct a function $f'_{\alpha_j} : \mathcal{T} \rightarrow \mathcal{N}$ that returns the next hop of a given MP2P $t_i \in \mathcal{T}$ in a node $\alpha_j \in \mathcal{N}$.

Let us assume that $g : \mathcal{P} \rightarrow \mathcal{T}$ is a mapping from one LSP in \mathcal{P} to one MP2P in \mathcal{T} . The problem is translated to find the best functions g and f'_{α_j} with the smallest domain such that the forwarding imposed by f_{α_j} is not altered, i.e., $f'_{\alpha_j}(g(p_i)) = f_{\alpha_j}(p_i)$ for all $p_i \in \mathcal{P}$ and all $\alpha_j \in \mathcal{N}$.

Some interesting properties of this formulation are listed below.

- Since the binary relationship g^{-1} creates a partition over the set \mathcal{P} , an LSP (identifier) just belongs to one MP2P connection.
- Because \mathcal{P} is partitioned (previous item) and f' is a function, the set of links corresponding to the LSPs in $g^{-1}(t_k)$ (for any MP2P connection $t_k \in \mathcal{T}$) form an *inverted tree*.

It is clear that for a single egress LSR $e \in \mathcal{N}$ there may be one or more MP2P trees; let $T_e \subseteq \mathcal{T}$ be the set MP2P trees with the same egress LSR e . To simplify notation, let N_{p_i} and N_{t_k} be the set of nodes used by path p_i and tree t_k respectively.

With these MP2P tree structures the label assignment is performed as follows.³ If β and γ are two consecutive LSRs in a tree t_k —i.e., $f'_\beta(t_k) = \gamma$ —and $f'^{-1}_\beta(t_k) = \{\alpha_0, \alpha_1, \dots, \alpha_k\}$, $k \geq 1$, is the set of LSRs whose next hop is β , then LSR γ may query β for a *unique* label L to forward packets for the LSPs going through α_i , $0 \leq i \leq k$. In this way, LSR β will map many incoming packets marked with different labels, coming from α_i , to the same outgoing label L and, hence, LSR γ will receive all of them using the same label. Therefore, the number of labels used (and NHLFEs as well) in γ is reduced from k to 1 for the LSP routes in $g^{-1}(t_k)$.

A. The Merging Problem

Since an MP2P connection is formed by its merged LSP routes, the main problem for creating it is to select LSP routes such that: a) there is *at most* one common path between any pair of LSP routes and, b) the common path (if it exists) *ends* in the egress LSR.

The configuration in Fig. 1 is an example that shows the MERGING PROBLEM. For this configuration, at least 2 MP2P connections must be created for the 3 LSPs shown, since LSR Nx is crossed by two LSPs: LSP B and LSP C—i.e., this prevents us to create a single tree with these 3 LSPs.

Depending on how the LSPs are grouped for forming MP2P connections, the label space may be reduced by more or less. For the example in Fig. 1, by merging LSP A with LSP C into a single MP2P tree and, in the other part, leaving LSP B alone the total number of labels⁴ is reduced by 1, while by merging LSPs

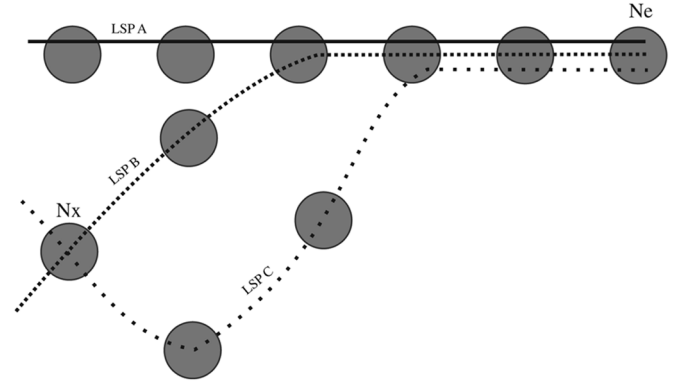


Fig. 1. MP2P Scenario with 3 LSPs with the same egress LSR $N1$. There may exist more than one way to perform merging since LSR n is crossed by two LSPs (LSP B and LSP C).

A with B and leaving LSP C alone, the total number of labels is reduced by 2.

B. Previous Work on MP2P

As stated in Section I.C, we need to compute the optimal LASPARED using MP2P. Consequently, we analyze and summarize in this subsection the most relevant previous work on MP2P label space reduction.

Applegate and Thorup propose in [3] an algorithm that builds $N + M$ MP2P trees, where N and M refer to the number of LSRs and links respectively in a given network. They affirm that each LSR uses at most $N + M$ labels since they bind the number of MP2P trees build to $N + M$. However, this is not clear considering that the merging LSRs of an MP2P tree must count 2 different NHLFEs for 2 different incoming interfaces of an LSR even if the incoming labels are the same (the reader can refer to Section 3.14 of [8]). Their bound is based on an optimization model that allows *rerouting* and assures the QoS level by assigning the same previously allocated bandwidth (but other QoS parameters are left out). Since we do not consider rerouting, the computed bound will not be taken into account.

Bhatnagar, Ganguly and Nath in [5] (and similar works like [14] and [18] as well) consider the LASPARED problem and they propose an algorithm to solve it by creating MP2P trees. In their work, their aim is to *minimize the number of MP2P trees* created for a given set of pre-computed LSPs routes. The previously described example in Fig. 1 shows that their goal may be *weak* for some configurations since there may exist 2 solutions with the same number of MP2P trees for a given set of LSP routes, but differing in the number of labels used. This weakness can be regarded in [4] as well, in which *et al.* proposed a Zero-One Integer Programming model with the same objective.

Although Bhatnagar's and Saito's work outlined the importance of reducing the label space *directly*, to the best of our knowledge, a solution for the MERGING PROBLEM, minimizing the number of labels using MP2P trees, did not exist.

C. MP2P Performance Overview

Two drawbacks are discussed in this subsection. The first drawback, briefly mentioned in Section II.A, is a consequence

³At time of publication, the IETF has been discussing a proper extension of RSVP-TE for signaling MP2P LSPs. The reader is addressed to [17] for this matter.

⁴Taking into account that LSR Ne does not receive labeled packets due to the IETF recommendation: *penultimate hop popping* in Section 3.16 of [8]

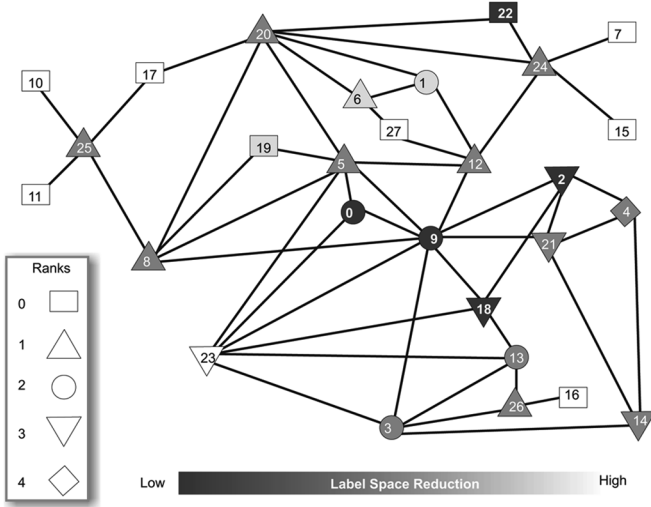


Fig. 2. Simplified version of the Australian Rocketfuel ISP topology. Nodes ranked according to closest egress proximity and colored according to level of label reduction using MP2P.

of the MPLS forwarding mechanism: *the label space of an LSR cannot be reduced using LSPs with different egress nodes.*

The second drawback is related to the treelike shape of MP2P connections:

Proposition 1 (MP2P Biased Reduction): The greatest percentage of reduced labels by MP2P in a network are located in LSRs near an egress LSRs.

In order to corroborate this statement, we carried out the following experiment.

A reduced version of the Australian ISP topology discovered by the Rocketfuel engine, shown in Fig. 2, was used. The simplified version has 28 nodes, each one corresponding to a different location in Australia.⁵ The nine nodes having the lowest connectivity degree were selected as edge routers.⁶ A set of 500 different LSPs were routed between the edge routers. The minimum number of labels using MP2P was computed. Throughout this work, we classify the relative label space reduction of a node in one of four ranges: [0%,25%]; [25%,50%]; [50%,75%) and [75%,100%). In the figure, the color of the nodes indicate its range of label space reduction using MP2P. At the same time, we ranked the nodes according to the minimum number of hops to the closest egress: zero for egress, one for egress' neighbors, and so on. As seen in the figure, the shape of the LSRs corresponds to their rank. Table I shows the average label space reduction percentage for each node rank in the example.

Finally, we computed the correlation coefficient between the nodes's rank and the percentage of the total number of labels reduced. The correlation is large negative (close to -0.8), confirming our previous statement. This means that the LSRs at the core of the network will not reduce their label space as much.

In the next section we recall from [19] a way of easing this asymmetry by stacking one label, or tunneling. Moreover, it will

⁵A set of nodes belonging to one location were mixed as a single node.

⁶We also performed experiments with randomly selected edge routers. The conclusion follows equally if the average hop distance to *all* the egress is considered instead. In the interest of space and non-redundancy, we do not include them in the article.

TABLE I
AVERAGE OF THE LABEL SPACE REDUCTION PERCENTAGE FOR EVERY RANK

Rank	Num. of Nodes	Avg. LaSpaRed
0	9	88.84%
1	8	44.39%
2	5	33.29%
3	5	26.70%
4	1	38.88%

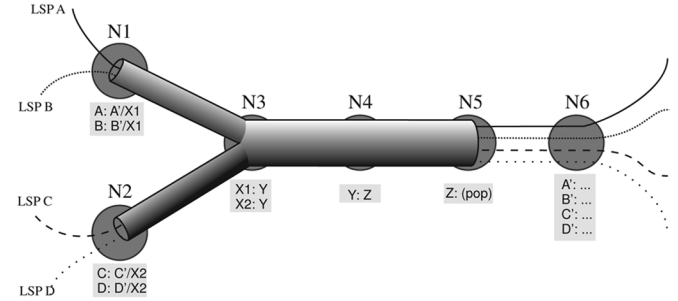


Fig. 3. AMT tree for 4 LSPs.

be shown later that our solution reduces the number of labels used even more when compared with MP2P.

III. ASYMMETRIC MERGED TUNNELING OR MP2P BY STACKING ONE LABEL

The notation used in Section II is preserved in this section. Let $p_k \in \mathcal{P}$ be an index of a path and let $N_{p_k} = \{\alpha_0^{(p_k)}, \alpha_1^{(p_k)}, \dots, \alpha_N^{(p_k)}\}$ be the set of LSRs it traverses. By pushing *one new label* onto the packets stack of LSP p_k at LSR $\alpha_i^{(p_k)}$ and popping the stack of the same LSP at LSR $\alpha_{j-1}^{(p_k)}$, a tunnel $\hat{p}_k = \{\alpha_i^{(p_k)}, \alpha_{i+1}^{(p_k)}, \dots, \alpha_j^{(p_k)}\}$ from p_k is obtained. This tunnel, \hat{p}_k , may be merged with other tunnels that end in the same LSR, $\alpha_j^{(p_k)}$. Considering $\hat{\mathcal{P}} = \{\hat{p}_0, \hat{p}_1, \dots, \hat{p}_{n'}\}$ as the set of tunnels created from the routes in \mathcal{P} and the merging scheme previously explained in Section II, inverse trees not rooted at egress LSR, but in each last LSR of the tunnel, can be created. As a consequence, the number of feasible label mergings for the paths in $\hat{\mathcal{P}}$ might be greater than those in \mathcal{P} .

In order to differentiate it from MP2P trees, explained in the previous section, we call this type of inverse tree *Asymmetric Merged Tunnel*, or AMT. Fig. 3 presents an example of it.⁷ LSR $N1$ swaps and then pushes label $X1$ for packets marked with labels A and B (belonging to LSPs A and B respectively). In the same way, LSR $N2$ swaps and then pushes label $X2$ to packets marked with labels C and D . LSR $N3$ merges them by swapping their labels with the same label Y . LSR $N4$ swaps the label of the tunnel with Z . LSR $N5$ then pops the stack for packets marked with label Z .

An AMT may be seen as a way to create MP2P trees where the root LSR may *not* necessarily be the *egress* LSR of a given subset of LSPs. By eliminating the MP2P restriction of sharing the same egress LSR for all merged LSPs, we are able to create

⁷The shadow pipe collecting links in figures means that the inner LSPs use one label stacked on those links. However, we consider that the last link of a tunnel is the next one following the last shadowed link. The reason is simple: although packets have been popped in the last shadowed link in figures, all of them (regardless of their LSP) must be forwarded toward the same LSR.

several MP2P-like trees *anywhere* in the network. This observation leads us to the following proposition.

Proposition 2 (AMTs Worst-Case Reduction): For any network and any set of LSP routes, AMTs achieve at least the same reduction factor than MP2P.

Given the previous analysis, the proof is straightforward. Every LASPARED solution using only MP2P is also a solution in AMTs, if and only if, every MP2P tree branch stacks one label at their extreme LSRs.

The natural questions are: *a)* how much can it be gained using AMTs? and *b)* what is the trade-off? The trade-off can be seen as the average increment (overhead) of the header size due to stacking, slightly affecting the traffic in the network. Clearly, the average header size would fluctuate between 4 (one label) and 8 bytes (two labels) depending on the number and location of the AMTs. Quantifications of the gain and the trade-off of AMTs are the main concerns of this article and they will be shown through simulations in further sections.

A. Tunneling Drawbacks

The use of AMTs, or tunnels, creates a hierarchy of LSPs, similar to that mentioned in [20]. The reader must be aware of the following drawbacks of using hierarchies in MPLS, as it is briefly exposed in [21].

- Hierarchies in general requires multilayer Traffic Engineering (TE). In this context, TE capabilities may be impacted in a re-optimization process, if not correctly addressed.
- The ingress and egress nodes of a tunnel cannot be protected using a *local* protection scheme, such as Fast Reroute. Therefore, other protection schemes are required, hindering the recovery time.
- Globally, this requires configuring and managing more LSPs.

B. The Stacking Problem

To compute an optimal AMT solution, we not only face the previously described MERGING PROBLEM on MP2P, but also the STACKING PROBLEM presented in the following.

The main decision problem that needs to be tackled when reducing label spaces through stacking is to decide where to *start* and *end* AMTs, *viz.* computing the proper set $\hat{\mathcal{P}}$. Let us consider the LSP configuration shown in Fig. 4(a); this configuration has two feasible AMT solutions which can be seen in Fig. 4(b) and Fig. 4(c).

In this example, the solution at the top [Fig. 4(b)] builds a tunnel that makes LSRs in the network use a total of 15 labels, while the second solution [Fig. 4(c)] makes them use 14 labels.

The most closely related work to this type of reduction is presented by *et al.* in [2] (and similarly in [12] and [13]). They studied the trade-off between label space sizes and stack depth in some special network configurations. They focus in network configurations in which all LSRs are interconnected either: *a)* along a path, or *b)* along a tree. Comparing it with our contributions in this paper, our problem is more general since we face the problem of *defining* the AMT trees in the network, as mentioned before, while *et al.* assume that the trees are already given.

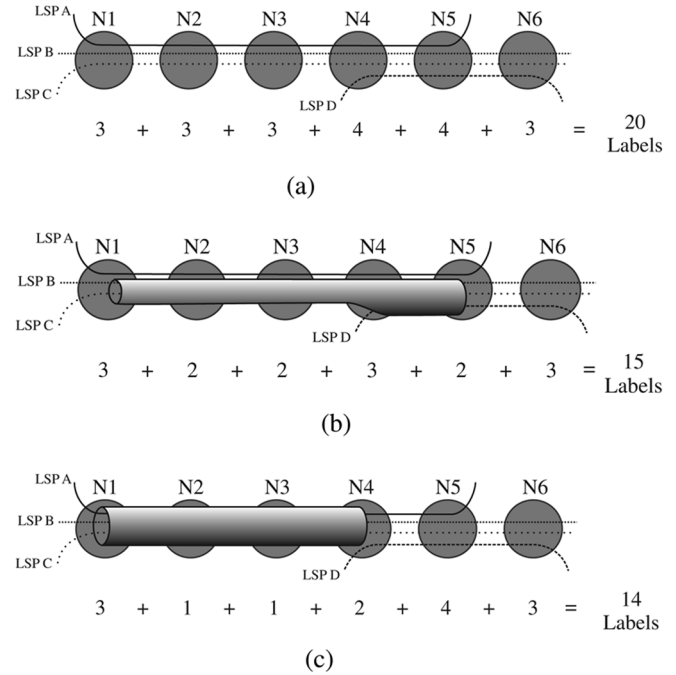


Fig. 4. Stacking problem example. (a) Paths and network configuration. (b) Suboptimal solution. (c) Optimal solution.

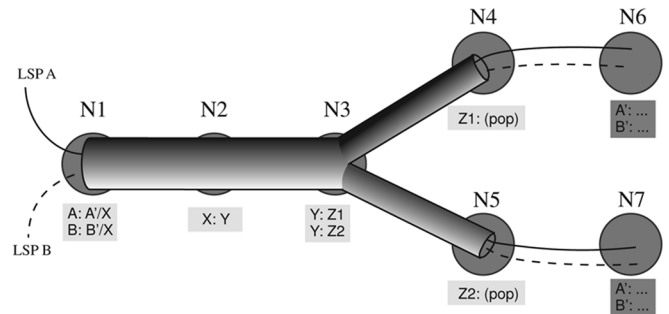


Fig. 5. Unfeasible Point-to-MultiPoint Tunnel.

C. Considering Multicast Connections

Stacking cannot be directly applied for reducing label spaces with multicast connections, best known in MPLS as Point-to-MultiPoint (P2MP) LSPs.⁸ Consider the example in Fig. 5, in which two P2MP LSPs traverse the same set of nodes.

When LSR *N3* receives a packet, it extracts the top label (e.g., *Y*) and looks for an NHLFE referring to the label. Based on the NHLFE information, the LSR duplicates the packet (one for *N4* and the other for *N5*). Before their respective forwarding, LSR *N3* swaps the top label in both outgoing packets with *Z1* and *Z2* respectively.

Notice that neither LSR *N4* nor LSR *N5* are able to swap the label underneath the top, neither for LSP *A* nor for LSP *B*. Therefore, at the end of the tunnel, the last LSRs (both LSRs *N6* and *N7*) get packets with the same label underneath the top. This situation is undesirable in MPLS since different LSRs may

⁸At the time of publication of this manuscript, the IETF has started working on an *upstream label assignment* method for LSP signaling [22] in order to properly setup P2MP LSPs and P2MP tunnels [23]. This approach seems to solve the P2MP label binding issue mentioned here.

not be able to agree in label bindings for a particular LSP: labels are *autonomically* selected [8].

Therefore, in order to reduce label spaces in P2MP connections, the connections have to be split into several Point-to-Point (P2P) connections at the branch points. In the example, each P2MP connection must be considered as three different P2P connections: $N1 \rightarrow N2 \rightarrow N3$, $N3 \rightarrow N4 \rightarrow N6$ and $N3 \rightarrow N5 \rightarrow N7$.

IV. THE BRUTE-FORCE MODEL

In this section, our first thought for modeling the LASPARED problem with the considered constraints is presented. We denote it as the *Brute-Force model* (or BF-model for short), since the computation takes the whole set of LSP routes and the complete network as input.

A. Parameters and Variables

Let \mathcal{T}, \mathcal{N} and \mathcal{P} be the set of indexes for: tunnels, LSRs, and LSPs path respectively. The symbol \mathbb{B} is used to denote the set $\{0, 1\}$. We use the following indexes in the model.

- $i, j, k \in \mathcal{N}$. Nodes in the network.
- $p \in \mathcal{P}$. An LSP path routed in the network.
- $t \in \mathcal{T}$. A tunnel.

Since the number of tunnels (or AMTs) is unknown *a priori*, the set \mathcal{T} must be dimensioned to a sufficiently large number.

Let $L_{(i,j)}^p \in \mathbb{B}$ be a (given) binary model parameter with a value of 1 if the link (i, j) forwards packets of the path p , and 0 otherwise. Since each link that a path uses makes an LSR use an additional label, the number of labels used in a network without AMTs equals $\sum_{p,(i,j)} L_{(i,j)}^p$, e.g., 16 in all the subfigures of Fig. 4.

The variables in the model are the following.

- $u_{(i,j)}^{p,t} \in \mathbb{B}$. Set to 1 when link (i, j) of path p is *stacked* in AMT t , 0 otherwise.
- $v_{(i,j)}^t \in \mathbb{B}$. Set to 1 when AMT t is using link (i, j) , 0 otherwise.
- $x_j^t \in \mathbb{B}$. Set to 1 when LSR j is the receiving LSR (i.e., the last LSR of a tunnel) for AMT t , 0 otherwise.
- $y_t^p \in \mathbb{B}$. Set to 1 when AMT t is stacking path p , 0 otherwise.

Henceforth, the membership symbol (\in) is going to be omitted for notation simplicity, unless it is strictly necessary.

When a tunnel t is used, note that the number of *labels used only by the tunnel* t is $\sum_{(i,j)} v_{(i,j)}^t$, e.g., in Fig. 4(b) this expression has a value of 5.

Note that the last link in the figure has no shade ($N5 \rightarrow N6$ in Fig. 4(b)), however we count it as part of the tunnel. We name this special link the *receiving link*, and the last LSR the *receiving LSR* (variable x_j^t) of the tunnel. Even though there is no assigned label in the receiving link of a tunnel (as shown in the figure), it has to be considered as part of it since all the “stacked” LSPs must use this link in order to comply with MPLS tag-forwarding.

Similarly, the expression $\sum_{p,(i,j)} u_{(i,j)}^{p,t}$ adds up to the number of *labels that are not being used* by LSPs p since they are being tunneled by t , e.g., in Fig. 4(b) this expression has a value of 12.

B. Optimization Model

With these parameters and variables defined, the optimization problem can be represented as follows: given a set of paths \mathcal{P} , the best reduction possible is computed by finding the values of $u_{(i,j)}^{p,t}$, such that they minimize the total number of labels, ε , in the network

$$\varepsilon = \sum_{p,(i,j)} L_{(i,j)}^p - \hat{\Delta}$$

where

$$\hat{\Delta} = \underbrace{\sum_{p,t} \left(\sum_{(i,j)} u_{(i,j)}^{p,t} - y_t^p \right)}_{\beta} - \underbrace{\sum_t \left(\sum_{(i,j)} v_{(i,j)}^t - x_j^t \right)}_{\alpha}.$$

The expression Δ corresponds to the number of *reduced labels* by the usage of AMTs. Within Δ , the expression outlined by α counts the number of used labels for *tunnels*; while the expression outlined by β counts the number of labels for *paths* that are “covered” by a tunnel. In other words, subexpression β counts the number of path hops that are stacked (hence, not used anymore). Because there is no reduction at the receiving LSR, the subtraction of $\sum_{p,t} y_t^p$ is done once in order to make Δ values accurate. In the same way, since tunnels do not have stacked labels in the receiving LSR link, x_j^t is subtracted from α . The relationship between these two subexpressions can be seen as: while subexpression β saves path labels, subexpression α pays for those savings by using tunnels. Hence, the difference between them, expression Δ , gives the overall number of labels reduced when AMTs are built in the MPLS network.

Overriding fixed values in previous formula, a Zero-One Programming model can be formulated as:

Maximize:

$$\hat{\Delta} = \sum_{p,t,(i,j)} u_{(i,j)}^{p,t} - \sum_{p,t} y_t^p - \sum_{t,(i,j)} v_{(i,j)}^t + \sum_{t,j} x_j^t \text{ subject to} \quad (1)$$

$$u_{(i,j)}^{p,t} - u_{(j,k)}^{p,t} + v_{(j,k)}^t \leq 1, \quad \forall i, j, k, p, t \quad (2)$$

$$2 \cdot y_t^p - \sum_{(i,j)} u_{(i,j)}^{p,t} \leq 0, \quad \forall p, t \quad (3)$$

$$\sum_i v_{(i,j)}^t - \sum_k v_{(j,k)}^t - x_j^t \leq 0, \quad \forall j, t \quad (4)$$

$$\sum_j x_j^t \leq 1, \quad \forall t \quad (5)$$

$$\sum_t u_{(i,j)}^{p,t} \leq 1, \quad \forall i, j, p \quad (6)$$

$$u_{(i,j)}^{p,t} - L_{(i,j)}^p \leq 0, \quad \forall i, j, p, t \quad (7)$$

$$u_{(i,j)}^{p,t} - v_{(i,j)}^t \leq 0, \quad \forall i, j, p, t \quad (8)$$

$$u_{(i,j)}^{p,t} - y_t^p \leq 0, \quad \forall i, j, p, t. \quad (9)$$

The objective function, (1), minimizes the number of labels used in the network by maximizing the number of labels reduced.

In (2), the model indicates to tunnel all LSPs that were tunneled previously in link (i, j) in the link (j, k) , only if a path is being tunneled in link (j, k) . By adding this equation over index j , it can be demonstrated that the model avoids P2MP tunnels, if the paths in \mathcal{P} are P2P connections.

It should be noted that since route paths are given, the model does not actually deal with LSP routing. Moreover, comparing it with most commonly used networking routing models, the ‘source node’ (first pushing LSR) and ‘destination node’ (receiving LSR) of a tunnel are not known, hence, they are part of the problem itself. In our model, (2) can be seen as a sort of node-link routing constraint in those types of routing models.

The remaining equations are easier to read. Since a tunnel requires at least 3 nodes, (3) does not allow solutions to tunnel less than 2 links per path. By taking (8) into (3), it is easy to see that the total length of a tunnel is at least 3 nodes.

(4) gives a definition of x_j^t by allowing it to take the value of 1 only when node j has an incoming link but no outgoing link. Considering this, (5) states that there can only be 1 receiving LSR and therefore only one tunnel per each t index.

(6) prevents two tunnels from stacking the same path over the same link. (7) restricts $u_{(i,j)}^{p,t}$ search space according to the values of parameter $L_{(i,j)}^p$, i.e., assures that only those links used by paths may be tunneled. (8) and (9) give a definition of $v_{(i,j)}^t$ and $y^{p,t}$ respectively assuming that $u_{(i,j)}^{p,t}$ is known.

C. Reduction Using MP2P

To measure the improvement between the reductions achieved when a single stacked label is used (AMT) compared to when there is no stacking solution (MP2P) (currently not found in the literature) the best reduction for MP2P must be computed. The model described in the previous subsection can be simplified for this special case.

To compute the reduction for MP2P, the model is relaxed so only the AMTs created are rooted at egress nodes of any LSP route. This is formalized by the following restriction over the model:

$$x_j^t \leq E_j, \quad E_j = \begin{cases} 1, & \text{if LSR } j \text{ is an egress LSR} \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

In addition, since MP2P computation involves only the LSPs that have the same egress LSR e in common, the set \mathcal{P} could be partitioned into several subsets \mathcal{P}_e (each of them containing only the paths ending at e) and we then apply the model for each subset \mathcal{P}_e , one set at a time. This may reduce the computational time required to solve the ILP model. Note that the objective function still holds for the reduction using MP2P.

V. THE DECOMPOSE AND MATCH FRAMEWORK

Looking at the BF-model, we see that it uses 4-dimensional variables. Thus, it is not hard to see that any ILP solver will need a long time and large space to give a solution for large networks; especially considering the quadratic space in terms of LSRs. Most of the times, this makes the model costly in terms of computational time and resources. Therefore, we analyzed the problem and propose in this section a fast and optimal way to do it: the *Decompose and Match framework*.

The framework described here to reduce the label space is divided in two parts: the *decomposition algorithm* and the *matching model*. The *matching model* is a path-based ILP model. Because of its path-based nature, the model needs a set of AMTs already defined (i.e., precomputed) as input, in addition to the LSP paths. With these parameters, the model matches the given AMTs with the given paths such that the maximum number of labels that can be saved is computed. The optimal matching may not include all the precomputed AMTs, since an LSP cannot be stacked by two AMTs at the same time. Clearly, computing all the feasible AMTs in a network would require an exponential algorithm, because computing all the feasible paths in a network is an exponential process. Instead, the *decomposition algorithm* computes a subset of all the feasible AMTs, containing the optimal solution for the problem.

A. The Decomposition Algorithm

As mentioned before, it should be pointed out that the optimal solution to the LASPARED problem uses a *subset* of the AMTs that will be computed in this phase by the decomposition algorithm. The smaller this feasible optimal set, the better the runtime performance.

Definition 1 (Segment): We consider a *segment* as a sequence of 2 or more network links (formally an ordered set of 2 or more link elements) denoting a route in a network using at least three consecutive nodes.

In this sense, all the given LSP paths are segments, but not the opposite. If s_k is a segment, $\|s_k\|$ represents the number of links that it comprises.

Definition 2 Converging Segment—COS: We say that a segment s_k is *Converging* considering a set of paths \mathcal{P} if $\forall p_i \in \mathcal{P}$, then $N_{p_i} \cap s_k = \emptyset$ or $s_k \subseteq N_{p_i}$.

In other words, let us consider a segment s_k and all the LSPs $\hat{P} \subseteq \mathcal{P}$ that use at least one of the links in s_k . A segment is said to be *converging* when all the LSPs \hat{P} use all the links in s_k , i.e., when all LSPs converge into (or, no LSPs diverges from) the path followed by s_k . For example, in Fig. 4(a), the segment $N1 \rightarrow N2 \rightarrow N3$ is a converging segment since LSPs A, B and C includes it and D is completely disjoint to it.

Definition 3 (Maximum Converging Segment—MACOS): Let s_k be a converging segment, let s_k^{\rightarrow} be a segment formed by the links in s_k plus one more downstream link, and s_k^{\leftarrow} be a segment formed by the links in s_k plus one more upstream link. Then, s_k is a *Maximum Converging Segment* if neither s_k^{\leftarrow} nor s_k^{\rightarrow} are converging segments.

For example, in Fig. 4(a), the segment $N1 \rightarrow N2 \rightarrow N3 \rightarrow N4$ is a maximum converging segment since $N1 \rightarrow N2 \rightarrow N3 \rightarrow N4 \rightarrow N5$ (s_k^{\rightarrow} in our definition) is not a converging segment (because LSP D contains a part of it).

To assure the correctness of the algorithm, an intermediate theorem has to be demonstrated.

Theorem 1 (MACOS Optimality): Let s_k be a MACOS considering the paths \mathcal{P} , and let \hat{s}_k be a segment formed by the links in s_k minus one link ($\hat{s}_k \subseteq s_k$), then either a tunnel constructed following the links of \hat{s}_k is not optimum, or \hat{s}_k is another MACOS.

Proof: Let L and \hat{L} be the set of indexes of all the LSPs that are forwarded through the links in s_k and \hat{s}_k respectively.

Let us consider the relationship between L and \hat{L} . On the one hand, if $L \neq \hat{L}$ then our proof concludes trivially since \hat{s}_k is another MACOS. On the other hand, assuming $L = \hat{L}$, \hat{s}_k is not an MACOS (by definition). In this case, the number of labels reduced with a tunnel following the links of \hat{s}_k is $(\|L\| - 1) \cdot \|\hat{s}_k\|$, since $L = \hat{L}$. Similarly, the number of labels that can be reduced using s_k is $(\|L\| - 1) \cdot \|s_k\|$. This reduction is greater than the reduction offered by \hat{s}_k , because $\|\hat{s}_k\| < \|s_k\|$. Since the LSPs contained in \hat{s}_k are exactly the same as those in s_k (MACOS definition), we conclude that constructing a tunnel following the links of \hat{s}_k is not optimal. ■

The above theorem extends our space of optimal solutions.

Corollary 1 (Joint MACOS Optimality): Let $\mathcal{S} = \{s_0, s_1, \dots, s_k\}$ be the set of all MACOS found for a set of LSP routes \mathcal{P} . Let \mathcal{S}^* be the set of all the segments resulting from joining two or more consecutive MACOS in \mathcal{S} . Let x be a segment not in \mathcal{S}^* , i.e., $x \notin \mathcal{S}^*$. Then, a tunnel following the links in x is not optimal.

The proof follows the same sequence as the previous one. The previous corollary leads us to consider consecutive MACOS as part of our optimal solution. So far, we have proved how to discard non-optimal solutions. As a consequence, the resulting search space has been shrunk to combinations of MACOS, whose computation is shown shortly.

Since two paths might be intersected in more than one segment, an ordinary intersection operation between two paths may lead to non-existent routes. Therefore, it is necessary to define operators that compute, not a single non-existent route, but a set of existent segments shared between two paths. We define a multi-intersection binary operator ($\dot{\cap}$) such that if $s_i \dot{\cap} s_j = \{s'_0, s'_1, \dots, s'_n\}$, then each s'_i represents an ordinal intersection, i.e., a consecutive sequence of links forming a path inside both s_i and s_j paths (or segments, in general). Similarly, we define a multi-difference binary operator ($\dot{-}$) such that if $s_i \dot{-} s_j = \{s'_0, s'_1, \dots, s'_n\}$, then each s'_i represents a consecutive sequence of links forming a path for one of the two LSP routes s_i or s_j , but never both.

With this new definition of the intersection and difference operator between segments, we are ready to propose and demonstrate an efficient way to compute the claimed set of AMTs:

Theorem 2: The optimal solutions expressed by MACOS Optimality (theorem 1) and Joint MACOS Optimality (corollary 1) lemmas are computed by:

$$\begin{aligned} \forall p_i, p_j \in \mathcal{P}, p_i \neq p_j, s'_k \in (p_i \dot{\cap} p_j), \|s'_k\| \geq 2 \\ \rightarrow s'_k \in \mathcal{S}' \end{aligned} \quad (11)$$

$$\forall s'_i \in \mathcal{S}', \rightarrow s'_i \in \mathcal{S} \quad (12)$$

$$\begin{aligned} \forall s'_i, s'_j \in \mathcal{S}', s_k \in (s'_i \dot{-} s'_j), \|s_k\| \geq 2 \\ \rightarrow s_k \in \mathcal{S}. \end{aligned} \quad (13)$$

Proof: Let s_k be a MACOS (or a set of consecutive MACOS) and (a, b) a link that either follows or precedes s_k . Let $L = \{l_0, l_1, \dots, l_n\}$, with $n \geq 2$, be a set of LSPs going through all links in s_k . Since s_k is a MACOS, then the LSP routes forwarded by link (a, b) must be $L' \neq L$. At this point we have to consider two cases: at least one LSP diverges in (a, b) , i.e., $L' \subset L$, or at least one LSP is added in (a, b) , i.e., $L' \supset L$. A third case is when both of the above cases happen

simultaneously and it can be demonstrated using Case 1 or Case 2.

Case 1— $L' \subset L$. Let l_i be one of the LSP routes that diverges, i.e., $l_i \in L - L'$. Then, it is clear that $\forall l_j \in L, l_i \cap l_j = x$, in this case (11) and (12) compute the value.

Case 2— $L' \supset L$. Let l_i be one of the LSP routes that is added at (a, b) , i.e., $l_i \in L' - L$. Let l_j be one of the LSP routes that goes through links in s_k , i.e., $l_j \in L$. Then, s_k can be found in one of the $l_j \dot{-} l_i$ results, in this case (11) and (13) compute the value.

Up to this point, we have computed a set of P2P segments that could be feasible optimal solutions. To extend the P2P segments into AMTs, we need to make combinations of these segments taking into account that the combined segments: a) must have the same end node and, b) the segments may be intersected at most in one place—the same conditions for MP2P trees. This may be done with the following recursive formula (assuming initially $\mathcal{T} \leftarrow \mathcal{S}$):

$$\begin{aligned} \forall t_i, t_j \in \mathcal{T}, \sup t_i = \sup t_j, \|t_i \dot{\cap} t_j\| = 1 \\ \rightarrow (t_i \cup t_j) \in \mathcal{T}. \end{aligned} \quad (14)$$

Since routes are considered as ordered sets, the expression $\sup t_i = \sup t_j$ means “if t_i and t_j end at the same place”. Due to this last equation, the algorithm runs in exponential-time in terms of the number of segments. However, it is possible to use a polynomial time algorithm if the decision of which P2P segments can be merged as AMTs is left to the ILP solver. Our simulations showed that the exponential-algorithm is the best suitable for the framework, since the number of combinations it has to perform in the last step is restricted only to MACOS ending in the same LSR.

B. The LSP-AMT Matching Model

Assuming that a set of AMTs has been computed considering a set of given LSP routes, the main issue is: *which subset of non-interfering AMTs should be used in order to achieve the best reduction? Which LSP routes should be tunneled on each AMT?* In other words, the problem has been simplified so we need to find only the best non-interfering *matchings* between pairs of LSPs and AMTs which reduce, as much as possible, the number of labels used. In this section we solve this question by proposing a Zero-One Integer Programming model, called the *LSP-AMT Matching* model (LAM-model).

The list of indexes of the model is:

- $e \in \mathcal{E}$. A link in the network;
- $p \in \mathcal{P}$. A given LSP path;
- $t \in \mathcal{T}$. A pre-computed AMT by the *decomposition algorithm*.

The model uses the following list of parameters.

- $L_{t,e} \in \mathbb{B}$. Set to 1 if link e is used by the feasible pre-computed tunnel t , 0 otherwise.
- $R_{p,t} \in \mathbb{N}^{+0}$. A natural number parameter set to the number of labels that can be reduced by using AMT t with LSP p . These values can be easily found as the number of common links between them minus one.⁹ Note that if tunnel t cannot reduce the label space for any LSR in the path p , $R_{p,t} = 0$.

⁹The constant value of one (1) must be subtracted because the last hop does not count in the reduction

In order to simplify notation, let us assume that $L_t^* = \sum_e L_{t,e} - 1$.

The model uses two decision variables.

- $x_{p,t} \in \mathbb{B}$. Set to 1 when the path p is used in AMT t , 0 otherwise.
- $z_t \in \mathbb{B}$. Set to 1 when AMT t is used for any path in the network, 0 otherwise.

The model is formulated as follows.

Maximize:

$$\hat{\Delta} = \underbrace{\sum_{p,t} (R_{p,t} \cdot x_{p,t})}_{\beta} - \underbrace{\sum_t L_t^* \cdot z_t}_{\alpha} \quad (15)$$

subject to:

$$\sum_t L_{t,e} \cdot x_{p,t} \leq 1, \quad \forall e, p \quad (16)$$

$$x_{p,t} - z_t \leq 0, \quad \forall p, t. \quad (17)$$

The objective function, (15), computes the number of labels reduced. The equation has the same two subexpressions (*viz.* α and β) explained in the previous section.

The constraint imposed by (16) allows a path to be stacked using only one tunnel in every link. (17) assures that the quantity L_t^* is paid for every tunnel used, independently of how many LSPs are stacked with it.

C. About MP2P Label Space Reduction Using D&M

As we outlined in Section II.B, many authors working on MP2P reductions have addressed the importance of computing the optimal label space reduction using MP2P trees. Our last contribution in this paper concerns with computing this value using our framework.

In order to do this, a slight simplification must be made to the decomposition algorithm. The simplification consists on just considering a subset of LSPs and AMTs that end at the *same* LSR of the network. As stated in Section III.C, the reduction of MP2P can be computed separately in sets of egress LSRs; therefore the set of paths \mathcal{P} can be partitioned into disjoint subsets \mathcal{P}_d taking the egress LSR d of each path as the discerning criteria. Then, by taking directly (14) with an initial value $\mathcal{T} \leftarrow \mathcal{P}_d$ a set of MP2P trees are computed for destination d . The process must be repeated for each destination $d \in \mathcal{N}$.

D. D&M Framework Performance

It may seem that an optimization model that computes the AMTs directly from the LSP routes (the BF-model) is a bit more complicated to formulate but equally efficient in performance as the framework presented in this section (the D&M framework). However, this is not true. In this subsection we will compare the performance of the D&M framework with both the BF-model and the model proposed by *et al.* for MP2P label space reduction in [4].

In order to compare the solutions offered for the LASPARED problem, throughout this subsection we use the variables s , t , and n to denote the number of LSP routes, AMTs and LSRs in the network respectively.

In the work of *et al.* in [4], the number of variables used is $O(tn^3)$ because: *a)* their model needs to set the links used for an MP2P and *b)* in order to make a fair comparison, we model all egress LSRs at the same time. Equation number 3 in Section III.B of [4] raises the number of constraints to $O(stn^4)$ since the restriction holds for every link, ingress node, MP2P solution and egress node in our comparison scheme.

In the *BF-model*, to take into account the LSP routes, we need to consider variables indexed over all the network nodes and the different LSP routes. Moreover, an additional index is needed to control the relationship between the route's segment and its AMT. Therefore, the space used by the variables is $O(stn^2)$. Because the BF-model needs to preserve the stacked LSPs between links in an AMT, the BF-model needs a sort of three-node flow conservation restriction. This restriction increases the number of constraints to $O(stn^3)$.

The *LAM-model* presented in Section V.B is composed of two binary variables. One of them uses $O(st)$ binary variables and the other uses $O(t)$ binary variables. In addition, the number of restrictions used by the model is $s \cdot (l + t)$, where l stands for the number of links used in the network; usually $l \leq n^2$. This leads us to $O(st)$ in the number of constraints, assuming that the number of feasible AMTs is greater than the number of links used in the network. In the same way, the space of the input parameters is $(s+l) \cdot t \in O(st)$. Clearly the LAM-model is much better than both the BF-model and Saito's model in terms of space and consequently running time.

The complexity of the *decomposition algorithm* in Section V.A is the price that must be paid for the efficiency of the model and the framework in general. Despite this, our simulation experiments (see next section) were executed faster than those of the BF-model. We would like to point out that our framework ran entirely in approximately 5 seconds even for tests with networks with 32 LSRs and 256 LSPs, while the BF-model could only be run up to a 7-LSRs network with 10 LSPs in the same computer (Solaris with 1 GB of physical available memory) using the same solver (CPLEX). Saito's work also showed a good runtime performance, but not as good as ours. Moreover, it should be kept in mind that Saito's work achieves LASPARED by means of MP2Ps *only* (it does not use the stack) which decreases the possibilities of reducing the label space; as it will be seen in the next section. Although processing time is dependent on the computer system and solver characteristics, it should be pointed out that the D&M framework performance becomes appropriate for hourly or daily operations for an ordinary network operator.

VI. SIMULATION RESULTS

In this section we present a set of simulation results that aim at showing the benefit of using AMTs. In our simulations, we considered the following aspects:

About the Network Topology. The network topology used in this article is based on the Australian ISP topology discovered with the Rocketfuel engine [24], showed previously in Fig. 2. The topology used here differs from the original in that ours has one node for every different location, i.e., a node may represent a set of different interconnected nodes within the same physical location in the original topology. Our topology has 28 LSRs, in

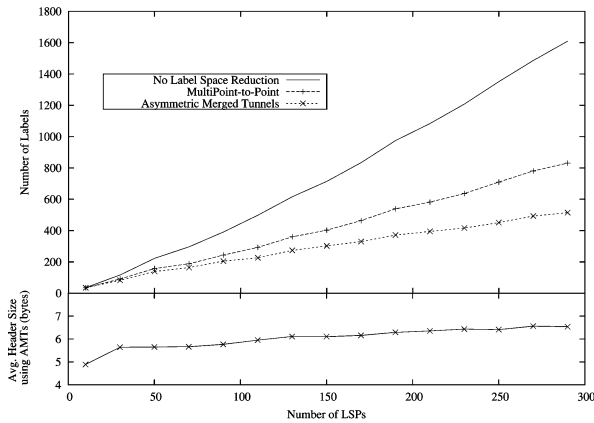


Fig. 6. Number of Used Labels and Overhead caused by Stacking in Function of the Number of LSPs.

which the nine with the least degree are always selected as edge LSRs.

About Routing. In order to ensure valid LSP paths over any generated network, the LSP routes were computed using a multi-objective evolutionary algorithm in which several QoS metrics were taken as objective functions for placing LSP routes [25]. This ensures that our LSPs routes may obey, to some degree, network operators' concerns. Explaining this routing algorithm is out of the scope of this paper, but the reader can see its specification in [25]. It should be pointed out that, since the routing algorithm is multi-objective [26], a set of possible routes may be computed for a single set of demands. In this case, the average of the reductions is taken into consideration.

About the Analyzed Metrics. The following metrics were considered in the simulation results.

- 1) *Number of labels used:* number of labels used when the label space reduction methods were used (MP2P and AMT).
- 2) *Average packet overhead with AMTs:* this may be measured as the number of stacked hops over the number of total hops in an AMT solution. Note that the average packet overhead of MP2P solutions is constant, since the stack sizes are not increased.

The rest of this section is divided as follows. The first subsection evaluates the gain of labels versus the overhead due to stacking. The second subsection shows that AMTs helps reduce more labels at the core nodes. While the analysis presented in these first two subsections is made varying the number of LSPs, the last subsection presents detailed results for a fixed high-number of LSPs.

A. Overall Reduction and Overhead

In this subsection, the minimum number of overall labels in the network is computed for different scenarios. Each scenario is comprised of different number of routed LSPs in the same topology.

In Fig. 6 we show at the top the number of labels used by each method when the number of LSPs is varied from 10 to 300. At

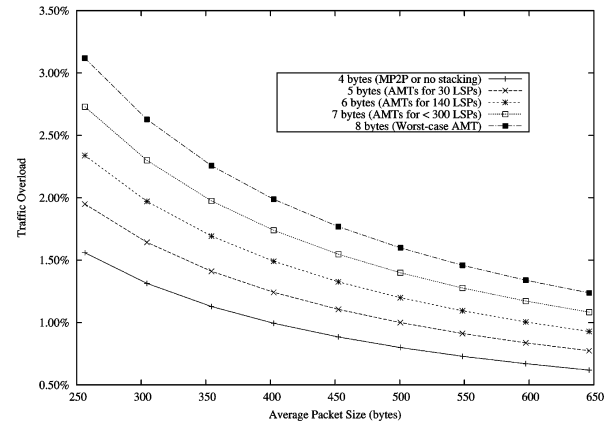


Fig. 7. Overload caused in the network traffic because of MPLS headers.

the same time, the bottom figure shows the average size of the incurred MPLS header.

The growing ratios are linear with the number of labels in all cases, but differ in their factor according to the LASPARED method used. As expected, AMTs use at least the same number of labels that MP2Ps. When the network load is high (300 LSPs), the label space reduction is 70.6% for AMTs, while MP2P is just 48.75%.

The cost of reducing 21.85% (70.6%–48.75%) the label space with AMTs is an overhead in the MPLS header. The average header size, as shown at the bottom of the figure, ranges from 5 bytes to 7 bytes in total (1 to 3 bytes of overhead, respectively) more than the normally used for MPLS tag-forwarding.

To analyze the overload caused in the traffic, we carried out a similar analysis to the one performed by *et al.* in [7]. We experimented with several traffic distributions. The traffic distributions consider three different sizes of payloads: 40 bytes, 520 bytes and 1500 bytes. The percentage of packets with payloads of 40 bytes can be 40%, 50%, and 60%, while the percentage of packets with payloads of 1500 bytes can be 2.5%, 12.5% and 22.5%. The percentage of packets with payloads of 520 bytes is set to the complement of the combination. For each traffic distribution, we computed the overload caused by the header size of MPLS. Fig. 7 shows these values.

It should be remarked that in the simulations carried out before, packet headers changed their size suddenly from 4 bytes to 8 bytes (and vice-versa) depending on the hop they took. This made the average header size vary from 5 to 7, as mentioned previously. However, we consider the case in which all LSPs are stacked at every point for a worst-case analysis. In other words, the worst-case analysis contemplates a fixed header size of 8 bytes at all hops for all LSPs, implying the maximum label space reduction possible using AMTs.

The typical traffic distribution of the traffic in the Internet can be considered as: 50%;37.5%;12.5% for payload sizes of 40, 520 and 1500 respectively, making an average payload size of 402.5. Compared to the worst-case, we conclude that the traffic overload is increased by 1%, if AMTs are used. In addition, even if the traffic distribution is changed to 60%;37.5%;2.50% (average payload size of 256.5), the traffic overload is increased by 1.56% only.

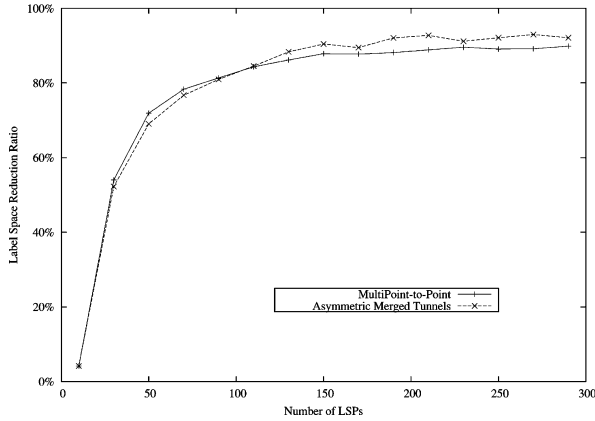


Fig. 8. Reduction Ratio for Nodes with Rank 0.

B. What Happens in the Core?

As shown in Section II.C, the larger percentage of labels reduced by MP2P is located in the LSRs close to the egress nodes. We show in this subsection that AMTs ease this drawback.

The simulation results previously presented are split into 5 different groups, each group considers only the LSRs of a particular rank. As mentioned before, in this article, the rank of an LSR is the minimum number of hops to the closest edge LSR in the topology. For every group, its average percentage of label space reduction for MP2P and AMT are computed. Since the fifth group is composed of only one node, only the first four groups are discussed and, hence, plotted between Fig. 8 and Fig. 11.

Fig. 8 shows the label space reduction ratio for edge LSRs; as expected, the gain of using AMTs with respect to MP2P is low. The gain improves as both the number of LSPs augments and the rank increases, as seen in Figs. 9–11.

To prove this observation, the correlation coefficient between the average gain obtained by using AMTs and the rank is computed. The coefficient was low positive (close to 0.3), which shows that the gain obtained by AMTs is not strongly related to the LSR ranks.

Sometimes some ranks may need to be sacrificed in order to obtain a better overall label space reduction using AMTs. For instance, between 110 and 190 LSPs, the solution obtaining the minimum label space requires that LSRs 18 and 23 (from rank three) create new tunnels that do not decrease their label space sizes. In fact, as it can be appreciated in Fig. 11, the reduction becomes negative (the number of labels for those LSRs is increased) for the AMT solutions in that range. However, it can be concluded that as the number of LSPs increases, the price that the latest ranks have to pay for an overall reduction is normalized.

This situation could be undesirable in scenarios in which the maximum number of labels used by any node in the network plays an important role, such as in AOLS. Even though the modifications in the ILPs to aim at this particular objective is small, its complete study goes beyond the purpose of this article, and it is left for future research.

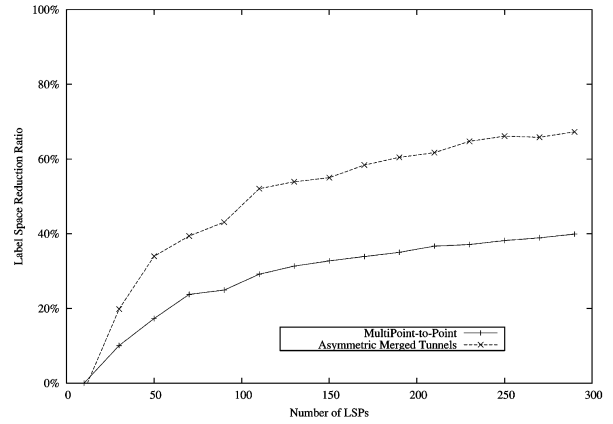


Fig. 9. Reduction Ratio for Nodes with Rank 1.

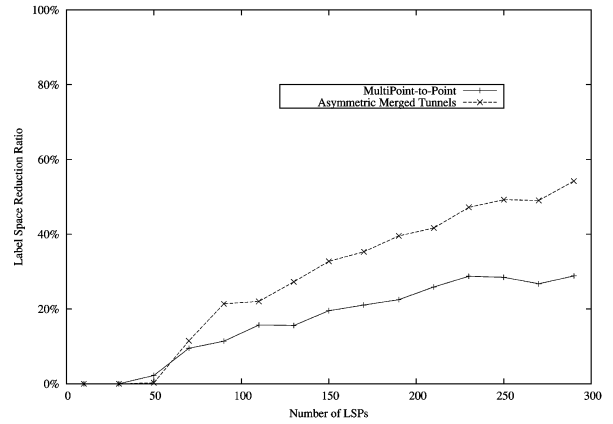


Fig. 10. Reduction Ratio for Nodes with Rank 2.

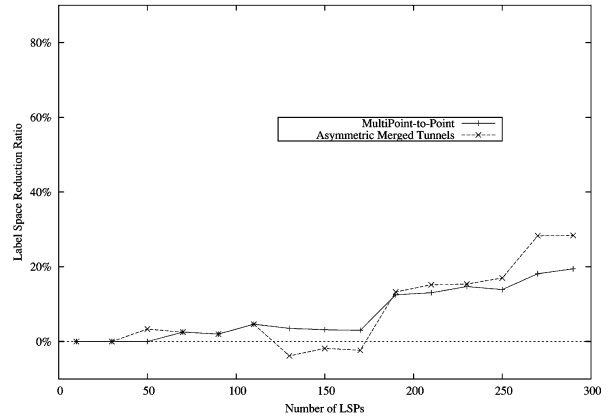


Fig. 11. Reduction Ratio for Nodes with Rank 3.

C. A Detailed Test

We performed a final test, this time with 500 LSPs routed in the same topology. In Fig. 12 we show the topology with the LSRs colored according to the label space reduction ratio in this test. Fig. 12 can be compared with Fig. 2 in order to appreciate the overall distribution of the label space reduction. The label space reduction of every LSR is shown in Table II.

Let us denote by $R_{(i)}$ the set of LSRs with rank i .

Furthermore, let us split the set $R_{(0)}$ in two disjoint sets $R'_{(0)}$ containing those LSRs with degree one (i.e., LSRs

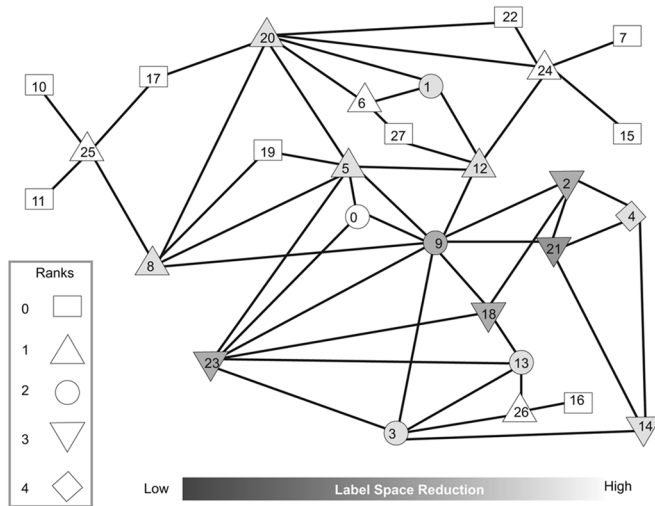


Fig. 12. Simplified version of the Australian Rocketfuel ISP topology. Nodes ranked according to closest egress proximity and colored according to the percentage of labels saved using AMT.

TABLE II
AMT VERSUS MP2P LABEL SPACE REDUCTION IN THE AUSTRALIAN
TOPOLOGY LOADED WITH 500 LSPs

Rank	Node	Total	MP2P	AMT	Absolute Improvement
0	10	52	98.08%	98.08%	0.00%
	11	52	98.08%	98.08%	0.00%
	15	52	98.08%	98.08%	0.00%
	16	58	98.08%	98.08%	0.00%
	17	139	79.86%	87.05%	7.19%
	19	100	73.00%	84.00%	11.00%
	22	120	78.33%	84.17%	5.83%
	27	99	77.77%	84.84%	7.07%
1	7	52	98.08%	98.08%	0.00%
	12	247	43.32%	70.45%	27.12%
	20	372	47.31%	69.89%	22.58%
	24	284	44.37%	75.70%	31.33%
	25	233	48.08%	77.25%	29.18%
	26	114	49.12%	85.96%	22.96%
	5	228	27.19%	52.19%	25.00%
	6	138	52.17%	76.09%	23.91%
2	8	241	43.57%	73.44%	29.88%
	0	41	17.07%	75.61%	58.53%
	1	135	51.85%	74.81%	22.96%
	13	74	41.89%	70.27%	12.5%
	3	82	42.68%	70.73%	28.05%
3	9	147	12.93%	25.85%	12.93%
	14	22	50.00%	54.54%	4.54%
	18	44	13.63%	43.18%	29.54%
	2	24	25.00%	37.50%	12.50%
	21	26	26.92%	34.62%	7.69%
4	23	78	17.95%	28.21%	10.25%
	4	18	38.89%	55.56%	16.67%
Total		3272	53.33%	70.67%	17.34%

0,11,15,16,7), and $R''_{(0)}$ containing the remaining ones (i.e., LSRs 17,19,22,27). We notice that the LSRs in $R'_{(0)}$ do not present any gain at all (0.00%) with the use of AMTs. This fact can be easily explained since all those LSRs have degree one. An LSR having degree one is never used as a transit LSR, i.e., all forwarded demands are either originated or ended there. The number of labels used for the demands *originated* in $R'_{(0)}$ cannot be decreased even though AMTs are used, since for every originated demand one different label is needed. Similarly, the number of labels used for the demands *ended* in $R'_{(0)}$

cannot be decreased using AMTs, since an AMT would always use the LSR's incoming links as the AMT receiving links. On the contrary, the LSRs in $R''_{(0)}$ experience a gain resulting from stacking of the demands that are traversing them.

Complementary, in order to corroborate our results, we carried out simulations over a set of 10 random network topologies generated accordingly to the work of *et al.* in [27]. Half of them containing 16 nodes and the other half 24 nodes. The network load was varied from 2 LSPs to 128 LSPs. All these simulations showed similar results, hence they are omitted from the article.

VII. CONCLUSIONS AND FURTHER STUDIES

In this paper we analyzed how the label space can be reduced by the addition of a single stacked label (AMT) in order to simplify network management and OPEX. For this, we proposed two different Zero-One Integer Linear optimization models, each of which finds the best way to reduce the label space by stacking only one additional label.

Simulation results show large improvements in the reduction of the number of labels using the stack (AMT) with respect to those found by traditional label merging schemes (MP2P). Furthermore, the analysis showed that by stacking only one extra label optimally (using AMT), the label space can be reduced by 70.6% compared to 48.75% when not using the label stack (using MP2P) in most of the networks scenarios. In addition, we also show that AMTs ease the biased reduction of MP2P towards LSRs close to the edge of the network.

To our knowledge, the label space reduction problem is a new open issue, therefore the following ideas have been left as *unresolved problems* in the area:

- to create an online algorithm for achieving online label space reductions;
- to develop a network signaling protocol based on RSVP-TE for label space reduction;
- to propose an approximation algorithm to reach a near-optimal solution to the label space reduction model;
- to study how All Optical Label Switching networks can work with this tunneling concept.

It is our aim to study the last two problems together in a near future.

ACKNOWLEDGMENT

The authors would like to thank Prof. J. Cavalcante de Oliveira, Drexel University, Philadelphia, PA, for all her comments. The authors also thank Dash Optimization for the free but limited usage of their (integer) linear problem solver—XPress^{MP} 2005B—through their Academic Partner Program.

REFERENCES

- [1] S. Pasqualini, A. Kirstadter, A. Iselt, R. Chahine, S. Verbrugge, D. Colle, M. Pickavet, and P. Demeester, "Influence of GMPLS on network providers' operational expenditures: A quantitative study," *IEEE Commun. Mag.*, vol. 43, no. 7, pp. 28–38, Jul. 2005.
- [2] A. Gupta, A. Kumar, and R. Rastogi, "Traveling with a pez dispenser (or, routing issues in MPLS)," *SIAM J. Comput.*, vol. 34, no. 2, pp. 453–474, 2005.
- [3] D. Applegate and M. Thorup, "Load optimal MPLS routing with N+M labels," in *Proc. IEEE INFOCOM 2003*, pp. 555–565.
- [4] H. Saito, Y. Miyao, and M. Yoshida, "Traffic engineering using multiple multipoint-to-point LSPs," in *Proc. IEEE INFOCOM 2000*, pp. 894–901.

- [5] S. Bhatnagar, S. Ganguly, and B. Nath, "Creating multipoint-to-point LSPs for traffic engineering," *IEEE Commun. Mag.*, vol. 43, no. 1, pp. 95–100, Jan. 2005.
- [6] F. Ramos, E. Kehayas, J. M. Martinez, R. Clavero, J. Marti, L. Stam-poulidis, D. Tsiokos, H. Avramopoulos, J. Zhang, P. V. Holm-Nielsen, N. Chi, P. Jeppesen, N. Yan, I. Tafur Monroy, A. M. J. Koonen, M. T. Hill, Y. Liu, H. J. S. Dorren, R. Van Caenegem, D. Colle, M. Pickavet, and B. Ripsosati, "IST-LASAGNE: Towards all-optical label swapping employing optical logic gates and optical flip-flops," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 10, pp. 2993–3011, Oct. 2005.
- [7] R. Van Caetiegem, D. Colle, M. Pickavet, and P. Demeester, "Benefits of label stripping compared to label swapping from the point of node dimensioning," *Photon. Netw. Commun. J.*, vol. 12, no. 3, pp. 227–244, Dec. 2006.
- [8] E. Rosen, A. Viswanathan, and R. Gallon, "Multiprotocol label switching architecture," IETF, RFC 3031, Jan. 2001.
- [9] E. Rosen, "MPLS label stack encoding," IETF, RFC 3032, Jan. 2001.
- [10] C. Neophytou and C. Phillips, "A scheme for the dynamic formation of robust multipoint to point LSPs," in *Proc. IEEE Consum. Commun. Netw. Conf. (CCNC 2004)*, Jan. 2004, pp. 251–255.
- [11] J. Marzo, L. Calle, C. Scoglio, and T. Anjali, "QoS on-line routing and MPLS multilevel protection: A survey," *IEEE Commun. Mag.*, vol. 41, no. 10, pp. 126–132, Oct. 2004.
- [12] A. Gupta, A. Kumar, and R. Rastogi, "Exploring the trade-off between label size and stack depth in MPLS routing," in *Proc. IEEE INFOCOM 2003*, pp. 544–554.
- [13] A. Gupta, A. Kumar, and M. Thorup, "Tree based MPLS routing," in *Proc. ACM Symp. Parallel Algorithms Architectures JSP/U 2003*, pp. 193–199.
- [14] S. Bhatnagar, S. Ganguly, and B. Nath, "Label space reduction in MultiPoint-to-Point LSPs for traffic engineering," in *Proc. IEEE Eur. Conf. Universal Multiservice Networks (ECUMN 2002)*, Apr. 2002, pp. 29–35.
- [15] D. Papadimitriou, E. Dotaro, and M. Vigoureux, "Ethernet layer 2 label switched paths (LSP)," in *Proc. Next Generation Internet Netw.*, Apr. 2005, pp. 620–621.
- [16] F. Solano, R. Fabregat, and J. Marzo, "On optimal computation of MPLS label binding for multipoint-to-point connections," *IEEE Trans. Commun.*, Jul. 2008, to be published.
- [17] S. Yasukawa, "Supporting multipoint-to-point label switched paths in multiprotocol label switching traffic engineering," IETF, Aug. 2007, draft-yasukawa-mpls-mp2p-rsvp-03.
- [18] S. Bhatnagar, S. Ganguly, and B. Nath, "Creating MultiPoint-to-Point LSPs for traffic engineering," in *Proc. IEEE Workshop High Performance Switching Routing (HPSR 2003)*, Jun. 2003, pp. 201–207.
- [19] F. Solano, R. Fabregat, and J. Marzo, "Full label space reduction in MPLS networks: Asymmetric merged tunneling," *IEEE Commun. Lett.*, vol. 9, no. 11, pp. 1021–1023, Nov. 2005.
- [20] K. Kompella and Y. Rekhter, "Label switched paths (LSP) hierarchy with generalized multi-protocol label switching (GMPLS) traffic engineering (TE)," IETF, RFC 4206, Oct. 2005.
- [21] S. Yasukawa, A. Parrel, and O. Komolafe, "An analysis of scaling issues in MPLS-TE backbone networks," IETF, Jul. 2007, draft-yasukawa-mpls-scaling-analysis-04.txt.
- [22] R. Aggarwal and J. L. Le Roux, "MPLS upstream label assignment for RSVP-TE," IETF, Mar. 2007, draft-ietf-mpls-rsvp-upstream-01.
- [23] S. Yasukawa, "Signaling requirements for point-to-multipoint traffic-engineering MPLS label switched paths (LSPs)," IETF, RFC 4461, Apr. 2006.
- [24] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring ISP topologies with Rocketfuel," *IEEE/ACM Trans. Netw.*, vol. 12, no. 1, pp. 2–16, Feb. 2004.
- [25] Y. Donoso, R. Fabregat, and J. Marzo, "A multi-objective optimization scheme for multicast routing: A multitree approach," *Telecommun. Syst. J.*, vol. 27, no. 2–4, pp. 229–251, Oct. 2004, Special issue on Networks.
- [26] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithm: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.
- [27] G. Siganos, M. Faloutsos, P. Faloutsos, and C. Faloutsos, "Power-laws and the AS-level Internet topology," *IEEE/ACM Trans. Netw.*, vol. 11, no. 4, pp. 514–524, Aug. 2003.



Fernando Solano (S'04) received the Ph.D. degree in the Broadband Communications and Distributed Systems group (BCDS), Universitat de Girona, Girona, Spain.

Currently, he is a Research Assistant at the Institute of Telecommunications in the Warsaw University of Technology. His research interests include optical transport networks design and analysis, all-optical label switching technologies, and mathematical modeling of network problems.



Thomas Stidsen received the Ph.D. degree in mathematics from the Technical University of Denmark, Copenhagen, in 2002.

Currently, he is an Associate Professor at the Institute of Informatics and Mathematical Modelling (IMM) in the Operations Research group, Technical University of Denmark. His research interests include combinatorial optimization methods for network protection optimization and network architecture optimization. He has co-authored a number of journal articles based on this research.



Ramon Fabregat received the Ph.D. degree in information technology from the Universitat de Girona, Girona, Spain, in 1998.

He is an Associate Professor of the Department of Electronics, Informatics and Automatics of the same university. His research interests are in the fields of management and performance evaluation of communication networks, network management based on intelligent agents, MPLS and GMPLS, and adaptive hypermedia systems. He coordinated the participation of the Broadband Communications

and Distributed Systems research group (BCDS) in the ADAPTPlan project (a national Spanish research project). He is a member of the Spanish Network of Excellence in MPLS/GMPLS networks. He has participated in the technical program committees of several conferences. He has co-authored the book entitled *Multi-Objective Optimization in Computer Networks* (CRC Press, 2007), and several papers published in international journals and conferences as well.



Jose L. Marzo received the Ph.D. degree in industrial engineering from Universitat de Girona, Girona, Spain, in 1997.

Currently, he is an Associate Professor at the Department of Electronics, Informatics and Automatics, Universitat de Girona. He leads the research group Broadband Communications and Distributed Systems (BCDS). His research interests are in the fields of management and performance evaluation of communication networks, network management based on intelligent agents, MPLS and GMPLS, and distributed simulation. His research group participates in several national Spanish research projects and the Spanish Network of Excellence in MPLS/GMPLS networks. He has co-authored several papers published in international journals and presented in leading international conferences.

Dr. Marzo is a member of the IEEE Communications Society. He has participated in technical program committees and chairing sessions of several international conferences. He serves in the editorial board of the *International Journal of Communications Systems*.